Matching road data of scales with an order of magnitude difference

Lüscher, Patrick ; Burghardt, Dirk ; Weibel, Robert

Abstract: Nowadays often a multiplicity of different spatial datasets exists for the same area. In order to use and maintain these datasets effectively they have to be integrated into a multiple representation database (MRDB). In an MRDB, the objects within the different datasets that model the same real-world phenomenon can be linked with each other. When integrating existing datasets these links have to be created by means of automatic matching techniques. In this paper we describe such a matching technique for road data of scales with an order of magnitude difference. The process starts by selecting possible candidates for roads and nodes using a buffer. The candidates are filtered using semantic, geometric and topological information. The remaining node candidates are compared by means of geometric measures, and 1 : 1 links between nodes are created. The node links are converted into road links by a shortest path algorithm. The method has been implemented and successfully tested with data at the scales of 1 : 25.000 and 1 : 200.000.

Matching road data of scales with an order of magnitude difference

Patrick Lüscher, Dirk Burghardt, and Robert Weibel
Department of Geography, University of Zürich, CH-8057 Zürich, Switzerland
E-mail: {luescher, burg, weibel}@geo.unizh.ch

ABSTRACT: Nowadays often a multiplicity of different spatial datasets exists for the same area. In order to use and maintain these datasets effectively they have to be integrated into a multiple representation database (MRDB). In an MRDB, the objects within the different datasets that model the same real-world phenomenon can be linked with each other. When integrating existing datasets these links have to be created by means of automatic matching techniques. In this paper we describe such a matching technique for road data of scales with an order of magnitude difference. The process starts by selecting possible candidates for roads and nodes using a buffer. The candidates are filtered using semantic, geometric and topological information. The remaining node candidates are compared by means of geometric measures, and 1 : 1 links between nodes are created. The node links are converted into road links by a shortest path algorithm. The method has been implemented and successfully tested with data at the scales of 1 : 25.000 and 1 : 200.000.

KEYWORDS: Data matching, multiple representation, map generalisation, spatial databases, road networks

1 Introduction

While in the past spatial analysts were constrained to the few datasets that were available, nowadays there exists often a variety of spatial datasets for a particular area that can be analysed together. Nevertheless, these datasets are usually stored separately of each other, and integration is carried out merely visually or by geometric overlay. This kind of data management is not efficient since updates have to be carried out on all datasets separately. Moreover, the full potential of common analysis cannot be exploited with simple integration methods.

A promising approach to tackle these problems is the integration of datasets into a multiple representation database (MRDB). In an MRDB, objects that model the same real-world phenomenon are linked with each other. One of the main challenges when integrating existing datasets into an MRDB is the automatic generation of links between corresponding objects by so-called matching algorithms.

In our research we have investigated the integration of road layers of the Swiss datasets VECTOR25 and VECTOR200 into an MRDB. VECTOR25 and VECTOR200 correspond in geometry and content to the Swiss National Maps 1 : 25.000 and 1 : 200.000, respectively. In this paper we present a matching algorithm which achieves excellent results in areas with low to medium population (and hence road) density.

2 Related Work

Besides data integration, spatial data matching can serve various other purposes such as the transfer of attribute data between two datasets, or the control and enhancement of geometric quality. Therefore, many approaches can be found in the literature concerning the matching of point, line and area type features. Here, we will constrain ourselves to describing the most influential approaches for road data.

One of the first attempts to matching road data is the work of Rosen and Saalfeld (1985). They aimed at associating survey maps of the United States Geological Survey (USGS) with maps of the Bureau of the Census. An iterative approach consisting of alternating matching and rubber sheeting has been chosen: In the matching part, nodes (i.e. road crossings) are associated. The rubber sheeting part, then, relies on nodes that have been associated in the matching part. It leads to a better geometrical correspondence of the two map sheets and thus allows the association of further nodes in the next iteration step. Linear (i.e. road) matches are created using associated nodes. Because the datasets involved were at similar scales, the approach is constrained to generating only 1 : 1 assignments between roads.

Devogele (1997) describes in his PhD thesis an approach for matching two datasets of slightly different scales: BD CARTO has mainly been created by digitizing 1 : 50.000 maps, while GEOROUTE is more detailed in urban areas. Due to the different level of detail, data models did not match: Typical differences were roundabouts of GEOROUTE that were represented as single points in BD CARTO, or complex road crossings that were collapsed to points in BD CARTO. Geometrically, displacements between corresponding roads were relatively small. The matching process consists of three stages:

1. *Creation of temporary road assignments*. The Hausdorff component of the road in the large-scale map to the road in the small-scale map serves as a measure for the distance between BD CARTO and GEOROUTE roads. A threshold for the Hausdorff component is iteratively reduced, in every iteration step unambiguous assignments are made temporarily.

2. *Correlation of nodes*. Two nodes can be linked when all of their respective incident roads were linked in step 1. If there are multiple candidates with partly linked roads, an n : 1 node assignment procedure is triggered. Thus, collapsed crossroads can be addressed.

3. *Final road assignment*. Shortest paths are calculated between matched nodes and assigned as road matchings.

The ideas of this algorithm were adopted by Mustière (2006) in order to conduct matching experiments for line networks between BD CARTO and BD TOPO, which is produced at a scale of about 1 : 25.000. His aim was to compare differences in modelling of data, and to study the feasibility of automated matching.

Walter and Fritsch (1999) developed a method for matching the German dataset ATKIS (Authoritative Topographic Cartographic Information System) to GDF (Geographic Data Files). Both datasets are produced at a scale of about 1 : 25.000. Unlike the previous approaches, Walter and Fritsch directly matched roads using linear properties without relying on previously matched nodes. In a first step, possible n : m road candidate pairs are generated through a process called *buffer growing*. Candidate pairs still may contain false assignments or they can be conflicting if a road object occurs in more than one candidate pair. They used an information theoretical approach to combine various geometric and topological measures such as line length, angle of the base line, etc. to an overall measure for the matching quality of each candidate pair. A hill climbing algorithm was then used to determine the best, unambiguous selection of the candidate pairs for final assignments.

The *buffer growing* approach to generate candidate pairs has been reused by several other authors, such as Mantel and Lipeck (2004). They compare the ratio of line lengths to determine optimal assignments: A threshold for the ratio is iteratively reduced; in every iteration step, those candidate pairs are matched having a ratio that exceeds the threshold and that don't conflict with another candidate pair. Zhang et al. (2005) use a similar approach to match roads of ATKIS base DLM with Teleatlas data. They use the node degree between two matching candidates supplementary to linear measures.

It is important to note that the approaches outlined above concentrate either on datasets that are of the same scale but different sources and therefore exhibit some inconsistencies, or that are of similar scale. In contrast, we were interested in the feasibility of matching data that have a larger difference in scale. Comparing scales 1 : 25.000 and 1 : 200.000, the strong generalization leads to several challenges when matching the two datasets: Firstly the more detailed dataset has a much denser network, and therefore many small road pieces have to be aggregated somehow and compared to one large road of the less detailed dataset. Secondly there are strong, locally constrained displacements (for example, refer to the road bend displayed in figure 8 left). Therefore the closest nodes do usually not represent corresponding road crossings. Thirdly, roads of the less detailed dataset are generally smoothed; hence corresponding roads are poorly comparable regarding shape measures.

In the remainder of the paper we will present a method that reaches high matching rates for datasets that are of scales 1 : 25.000 and 1 : 200.000. Nevertheless, a fully automated solution could not be achieved because of the complexity of the task, and our matching algorithm will to some extent rely on user checking and/or completion of results. Therefore, one important focus of our work was an appropriate visualization of matched data and user guidance to allow efficient matching sessions.

3 Matching Method

3.1 Overview

When matching data of different scales, usually the smaller scale is defined as reference dataset. For each object in the reference dataset those objects in the comparison dataset are to be found that, as a whole, correspond to the reference object (Dunkars 2003). This approach has also been followed in our work. The VECTOR200 road network is generally a subset of the VECTOR25 network: Out of the objects of VECTOR25, mainly those objects that are important for national and regional traffic routing are kept, while minor roads, agricultural roads, etc. drop out. Therefore, relations between VECTOR25 and VECTOR200 are normally of cardinality n : 1 and we constrained our approach to generating only n : 1 matches between roads. An extension to n : m associations is conceivable for instance by merging contiguous buffers with *buffer growing*. The process can be divided into four stages as follows:

I. Generation of candidate sets
Using a buffer around VECTOR200 nodes and roads, VECTOR25 candidate sets are generated for each node and each road of VECTOR200.

II. Matching of nodes
1 : 1 matches between nodes are created automatically. If no unambiguous correspondence can be found for a node, either because none of the candidate nodes

differs significantly from the other candidate nodes, or because no 1 : 1 assignment is possible, the situation is presented to the operator who has to match the node manually.

III. Matching of roads

Node assignments are automatically converted into road assignments. A shortest path algorithm is used for that purpose.

IV. Post processing

It is possible that not all roads can be matched automatically and correctly. Therefore, the operator has to control the assignments and complete them where needed.

In the following sections, the four stages of the process are explained in more detail.

3.2 Generation of candidate sets

Figure 1 illustrates the first stage of the process. The dashed boxes are labelled corresponding to the paragraph in which they are explained below.

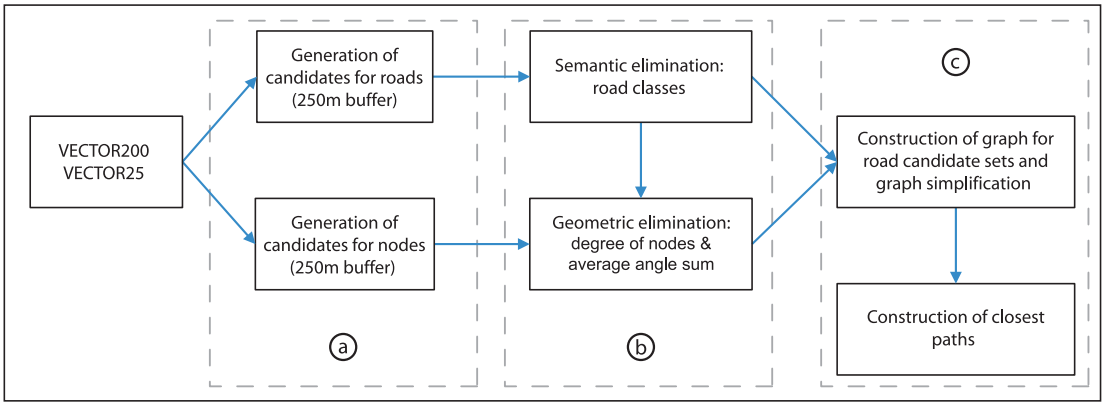

Figure 1. Generation of candidate sets.

a. Generation of preliminary candidate sets

Using a buffer of 250m width, a preliminary selection of candidates is generated for each road and each node of the VECTOR200 network. The buffer size has been determined empirically such that they include corresponding VECTOR25 roads in all cases. In the preliminary candidate sets many incorrect (non-corresponding) VECTOR25 roads can be found that have to be removed by subsequent filter mechanisms. Figure 4 left shows an example of VECTOR25 candidates after the buffer selection process.

b. Elimination of unlikely candidates

In this step, the preliminary selection of candidates is refined. To this end, additional criteria are defined that the candidates have to fulfil. When continuous measures such as Euclidean distance for point features or length difference and Hausdorff distance for linear features are used, usually thresholds are defined in order to decide whether a candidate is valid or not. Threshold values can be determined interactively by controlling results of the matching process, or by statistical evaluation of manually matched training areas. In our case, useful threshold values were derived from two manually matched training areas.

We used three additional criteria to eliminate incorrect candidates: For road candidates, the semantic criterion "object class" has been used, for node candidates a topological criterion "node degree" and a metric criterion "average angle sum".

Both VECTOR25 and VECTOR200 specify object classes for roads, but the classification is different, such that there exists no 1 : 1 correspondence between classes of the two datasets. Nevertheless, there are some classes that never occur together in corresponding road sets (e.g., a highway will never correspond to an agricultural road), while others are quite frequent. These correspondences have been encoded in a binary cross tabulation (table 1).

VEC25 / VEC200	1_Klass	2_Klass	3_Klass	4_Klass	5_Klass	6_Klass	Parkweg	Q_Klass
DurchgStr6	1	1	0	0	0	0	0	0
VerbindStr6	1	1	0	0	0	0	0	0
VerbindStr4	1	0	0	0	0	0	0	0
NebenStr3	1	1	1	0	0	0	0	1
Fahrstraess	0	1	1	1	0	0	0	1

Table 1. Compatibility of VECTOR25 / VECTOR200 object classes (0 = not compatible, 1 = compatible). Rows: VECTOR200 object classes. Columns: VECTOR25 object classes.

If, for example, a VECTOR200 road is of type "Durchgangsstrasse 6m", all VECTOR25 candidates which are not of type "1. Klass Strasse" or "2. Klass Strasse" can be eliminated.

The following criteria have been adopted to filter node candidates:

1. A VECTOR25 candidate node must have the same or a higher degree as the VECTOR200 reference node.
2. The roads that are incident to the candidate node and to the reference node are assigned to each other such that the angle sum between roads is minimal for the assignment (refer to figure 2). For valid candidate nodes, the average angle sum has to be smaller than 45°:

$$(\sum_{i=1}^{\kappa} \gamma_i / \kappa) \leq 45°$$

Where κ is the node degree of the reference node
γ_i are the angles between assigned roads

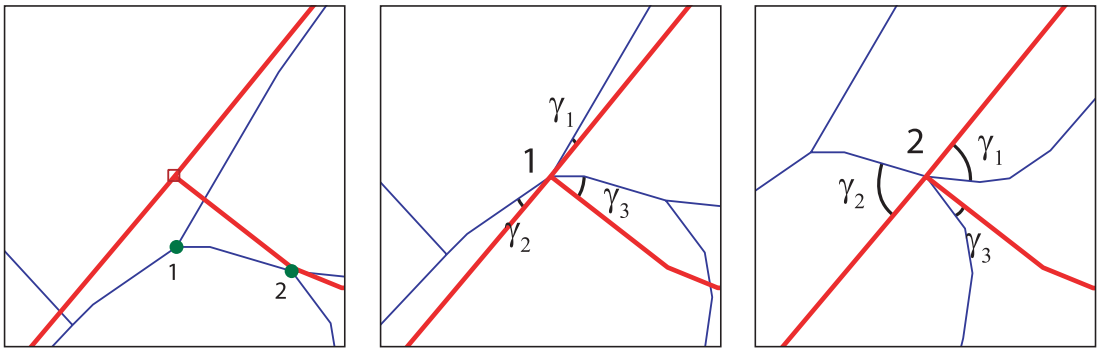

Figure 2. Comparison of angle sums.

c. Construction of graph structure and closest paths

Even after filtering by road classes there remain many incorrect candidate roads. To resolve this problem, Devogele (1997) proposed to calculate shortest paths between matched end nodes. We used a similar method based on a shortest path algorithm and the Hausdorff distance. The Hausdorff distance is a measure for the maximum distance between two lines. It can be calculated as the maximum of the two Hausdorff components (see figure 3 for an explanation). However, for matching linear data at different scales it is better to use only the Hausdorff component from the smaller scale dataset to the larger scale dataset (Devogele 1997). Therefore, we implemented an algorithm described in Hangouët (1995) to calculate the Hausdorff components $\Delta_{VECTOR25 \rightarrow VECTOR200}$.

$$\Delta_{A \rightarrow B} = \sup_{x \in A}(\inf_{y \in B} d(x, y))$$
$$\Delta_{B \rightarrow A} = \sup_{y \in B}(\inf_{x \in A} d(x, y))$$
$$\Delta_{Hausdorff} = \max(\Delta_{A \rightarrow B}, \Delta_{B \rightarrow A})$$

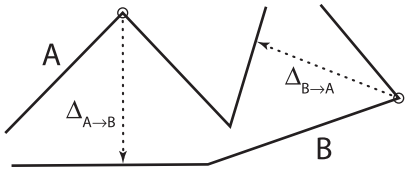

Figure 3. Calculation of the Hausdorff distance (after Hangouët 1995).

Now, instead of the path length we minimize the average Hausdorff component of a path in VECTOR25 to its VECTOR200 reference road. The result is a path which has the smallest offset from the VECTOR200 reference road and therefore is considered to be the set of candidates that constitute the most similar path. We termed this the "closest path".

Furthermore, the nodes don't need to be matched unambiguously – we rather calculate the set of all possible closest paths between all node candidates of two end nodes of a VECTOR200 reference road. VECTOR25 candidate roads which are not part of a closest path for a reference road are then removed from the candidate set of this reference node.

After this filtering procedure, the final candidate sets for the actual matching procedure remain. Figure 4 (left) shows an example for a candidate set before filtering, and figure 4 (right) the remaining candidates for the matching procedure.

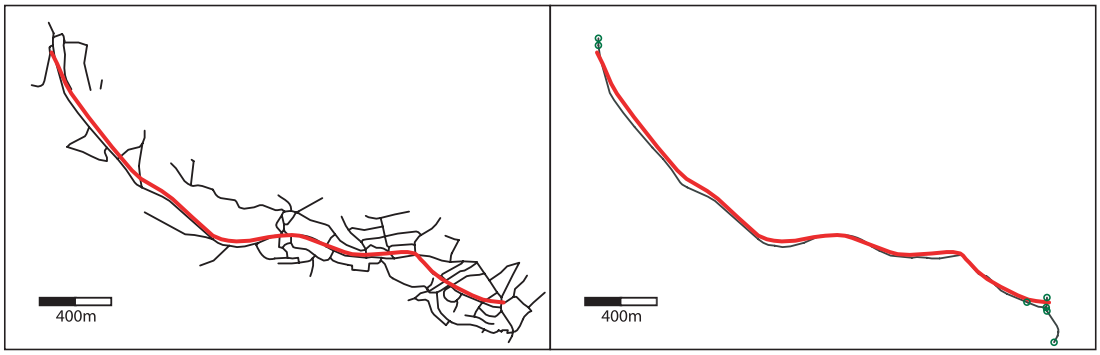

Figure 4. VECTOR200 reference road (red) and VECTOR25 candidates (black). Left: After candidate generation by buffers. Right: After filtering by closest paths. Green: Node candidates for which closest candidates have been calculated. VECTOR25/200 © 2007 swisstopo (BA071321).

3.3 Matching of nodes and lines

Even after filtering, a reference node usually has several candidate nodes. Unambiguous node matches are generated in an iterative process which we describe in this section.

The remaining node candidates are compared with respect to their distance to the reference node and the average angle sum. If a node candidate shows significantly smaller values than the other candidate nodes, it is considered to be the corresponding node and matched to the reference node. Remaining node candidates and their incident closest paths are removed. Through this step, the situation is simplified such that potentially additional nodes can be matched during the next iteration. We also use a line tracing algorithm for matched nodes to simplify the situation further.

If no new node matches could be generated during an iteration step, user interaction is needed. The user is guided to a VECTOR200 node that hasn't been matched so far. The candidate set for the node is visualized, from which the user has to select the correct corresponding node. He can also decide that the VECTOR200 node under investigation has no counterpart in VECTOR25. This happens when there are inconsistencies between the two datasets. Also, n : 1 assignments between nodes that occur when roundabouts or complex crossroads of VECTOR25 have collapsed to one single node in VECTOR200 cannot be automatically detected and have to be assigned interactively.

Once both end nodes for a VECTOR200 road are successfully assigned, the road itself can be matched. VECTOR25 roads that have to be linked with a VECTOR200 reference road are contained in the corresponding closest path between the corresponding VECTOR25 nodes. Therefore, this step runs fully automatedly.

3.4 Post processing

Several tools are provided to the user for post processing. They are integrated in a toolbox which is portrayed in figure 5. Tools (1) – (5) provide means to manually generate new matches or edit existing, incorrect matches. Tools (6) – (8) serve for user interaction in the node matching process described in the preceding section.

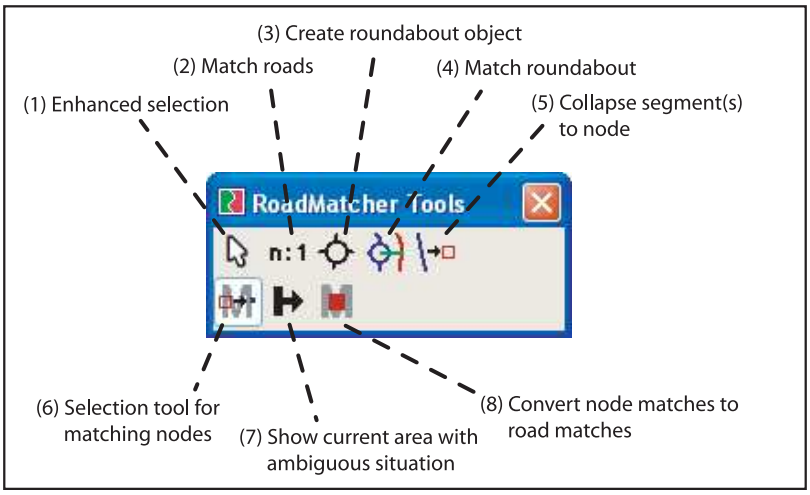

Figure 5. Toolbox of our matching application

4 Evaluation

4.1 Overview

The matching algorithm has been tested in two areas of size 10x10 km^2 each. The area "Pfäffikon" contains a dense network of small- and medium-sized towns and thus corresponds to the typical settlement structure of the Swiss Midlands region. The area "Winterthur" covers a city with 90'000 inhabitants.

Figures 6 and 7 show extracts for both areas. Figure 8 (left) shows a road loop. Large displacements have lead to significant positional differences between the two datasets. In the figure shown they range up to 200m. The algorithm can solve this situation correctly. Figure 8 (right) shows a situation where the matching algorithm cannot find a solution, because the two candidate nodes 1 and 2 are equally similar to the reference nodes with respect to the measures we use. Here, the user has to select one of the two nodes interactively.

Figure 6. Extract of area „Pfäffikon". Left: VECTOR25 (blue) superimposed with VECTOR200 (red). Right: Extracted VECTOR25 roads. VECTOR25/200 © 2007 swisstopo (BA071321).

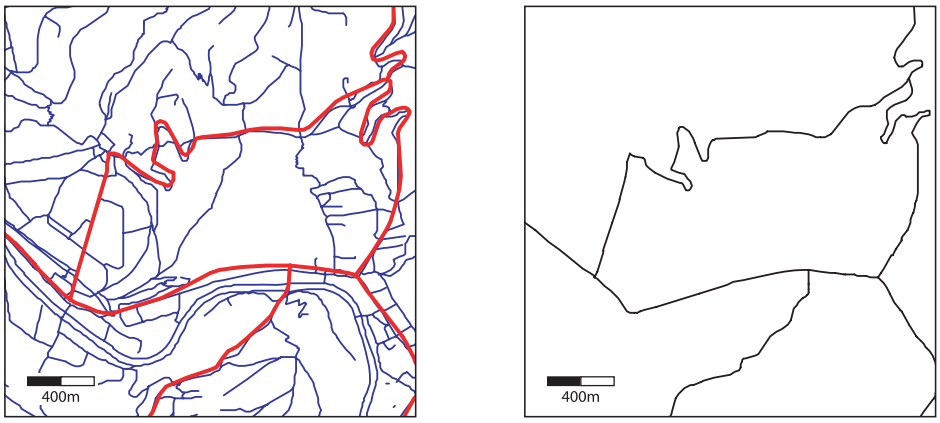

Figure 7. Extract of area „Winterthur“. Left: VECTOR25 (blue) superimposed with VECTOR200 (red). Right: Extracted VECTOR25 roads. VECTOR25/200 © 2007 swisstopo (BA071321).

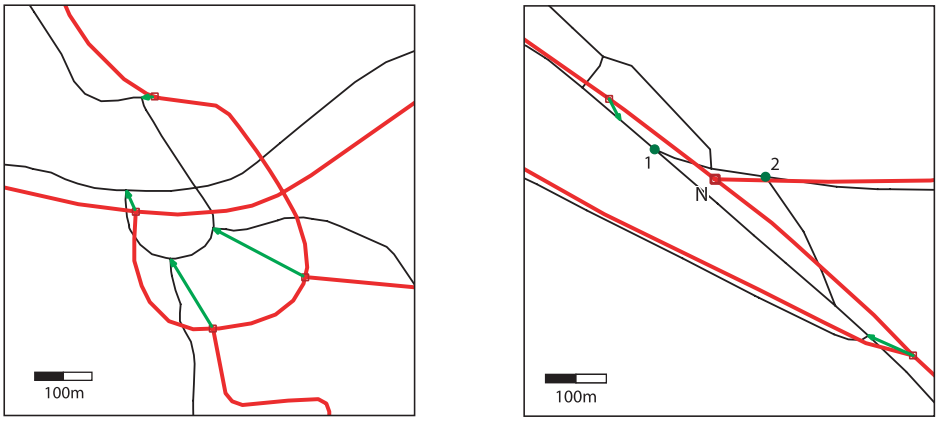

Figure 8. Left: Extracted road loop with node assignments. Right: Ambiguous situation with two candidate nodes, where user interaction is needed. VECTOR25/200 © 2007 swisstopo (BA071321).

4.2 Analysis of results and discussion

Table 2 shows the matching rates in both test areas for node matches and road matches. For the study area “Pfäffikon” 91.7% of nodes and 89.3% of roads could be matched automatically. Out of the nodes that had to be matched interactively, most were located at a border of the study area. For these nodes with degree 1 there were often a great many candidates. However, this edge effect becomes less important with larger areas. For the test area “Winterthur” matching rates are slightly lower with 84.0% for nodes and 79.9% for roads, because the urban situation is more complex.

In the test area “Pfäffikon”, no mistakes were found for the automatically generated matching, in the study area “Winterthur” all node matches were correct, but in one case an automatically generated road matching was faulty.

	Pfäffikon	Winterthur
Nodes of VECTOR200	97 (100%)	119 (100%)
matched automatically	89 (91.7%)	100 (84.0%)
matched interactively	5 (5.2%)	13 (10.9%)
no 1 : 1 correspondence	2 (2.1%)	5 (4.2%)
no correspondence (inconsistency)	1 (1.0%)	1 (0.9%)
Roads of VECTOR200	112 (100%)	144 (100%)
matched automatically	100 (89.3%)	115 (79.9%)
matched interactively	11 (9.8%)	28 (19.4%)
no correspondence (inconsistency)	1 (0.9%)	1 (0.7%)

Table 2. Matching rates of the automated matching process

Then again, results of the automated matching process were not satisfying in complex urban areas such as the inner city of Zurich. Partly the situations were inconsistent or they differed so strongly that even manually a matching could hardly be found. Also, the average angle sum is less efficient as a measure in urban areas since roads usually intersect perpendicularly. Because many of the streets are straight, an algorithm enhanced with line measures would achieve better results in urban areas. The employment of the measures "difference of road length" and "angle between base lines of roads" has been tested in less densely populated areas and unfortunately had a rather negative impact on the matching accuracy.

Therefore, we propose as extension of the current method to analyse first the region that has to be matched and characterize it as "suburban/rural" or "urban", respectively. The algorithm can then be parameterized accordingly.

5 Conclusion and future work

In this paper we presented a new set of methods for matching linear objects. The technique gains its major importance in the context of multi representation databases, which facilitate the update process and allow generating datasets of intermediate scales. In sharp contrast to existing work, we used datasets that were at largely dissimilar scales.
The approach that has been described determines for each road object of the smaller scale the corresponding objects of the larger scale. After generation of candidate nodes through a buffer operation, invalid candidates are gradually reduced by applying various techniques that use semantic, geometric and topological information. As a result, 1 : 1 node matches are generated that are extended to road matches by an extended shortest path algorithm. For an application in complex urban areas, additional line measures would have to be included. In addition, a situation analysis has to be developed beforehand.

Acknowledgments

Extract of swiss dataset VECTOR25/200, reproduced with permission of swisstopo (BA071321).

References

Cecconi, A. (2003): *Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping*. PhD Thesis, University of Zurich.

Devogele, T. (1997) : *Processus d'intégration et d'appariement de Bases de Données Géographiques Application à une base de données routières multi-échelles*. PhD thesis, Université de Versailles.

Dunkars, M. (2003): Matching of Datasets. In: *ScanGIS'2003 – The 9th Scandinavian Research Conference on Geographical Information Science – Proceedings*, Espoo, Finland, June 4–6, 2003, 67–78.

Gabay, Y. and Doytsher, Y. (1995): Automatic feature correction in merging of line maps. In: *ACSM/ASPRS '95 Annual Convention & Exposition Technical Papers*, Charlotte, NC, USA, February 27 – March 1, 1995, 2:191–199.

Hangouët, J.-F. (1995): Computation of the Hausdorff Distance Between plane Vector Polylines. In: *Auto-Carto12*, *ACSM/ASPRS Annual Convention & Exposition Technical Papers*, Charlotte, NC, USA, February 27 – March 1, 1995, 4:1–10.

Mantel, D. and Lipeck, U. (2004): Matching Cartographic Objects in Spatial Databases. In: *Proceedings of the XXth Congress of the ISPRS*, Istanbul, Turkey, July 12–23, 2004, Int. Archives of Photogrammetry, Remote Sensing and Spatial Inf. Sciences Vol. XXXV, Commission IV Papers, Part B4, 172–176.

Mustière, S. (2006): Results of experiments on automated matching of networks at different scales. *ISPRS Vol. XXXVI. ISPRS Workshop on Multiple Representation and Interoperability of Spatial Data*, Hannover, Germany, February 22–24, 2006.

Rosen, B. and Saalfeld, A. (1985): Match criteria for automatic alignment. In: *Digital representations of spatial knowledge: Auto-Carto 7 proceedings*, Washington, USA, 456–462.

Walter, V. and Fritsch, D. (1999): Matching spatial data sets: a statistical approach. In: *International Journal of Geographical Information Science*, **13** (5):445–473.

Yuan, S. and Tao, C. (1999): Development of conflation components. In: *Geoinformatics and Socioinformatics. Proceedings of Geoinformatics '99 and International Conference on Geoinformatics and Socioinformatics*, Ann Arbor, MI, USA, June 19–21, 1999, 1–13.

Zhang, M., Shi, W. and Meng, L. (2005): A generic matching algorithm for line networks of different resolutions. *8th ICA Workshop on Generalisation and Multiple Representation*, A Coruña, Spain, July 7–8, 2005.